2D Navigation with a Differential Wheeled Unmanned Ground Vehicle

Research · June 2017

DOI: 10.13140/RG.2.2.20876.16006

2 authors, including:

Olivier Gougeon
Polytechnique Montréal
7 PUBLICATIONS **9** CITATIONS

SEE PROFILE

2D Navigation with a Differential Wheeled Unmanned Ground Vehicle*

Olivier GOUGEON† & Mohammad H. BEHESHTI†

Abstract— **The attitude and position estimation problem for unmanned ground vehicles (UGV) is investigated by fusing measurements from an inertial measurement unit (IMU) and position sensors (incremental encoders). The sensor fusion approach is based on the extended Kalman filter (EKF) developed by [1], which uses the measurements from these sensors for the localization problem. Simulations are performed on a four-wheeled mobile robot by using a 2D kinematic model in the MATLAB/Simulink environment and the performance of the fusion algorithm is analyzed with the Husky platform [2]. The comparison between the simulation and experimental results clearly shows that limiting the problem to 2D is not sufficient for real-life applications. A more general 3D solution is then suggested.**

I. Introduction

A. Context

UGV are widely used all across the world. These autonomous robots can accomplish many different tasks such as delivering, mapping, monitoring, etc. For example, researchers of the University of Chile are developing algorithms so UGV can autonomously explore dangerous mines - therefore preventing humans from having to be exposed to these risky environments [3].

B. Problem Description

To operate in unknown environments, UGV need to rely on navigation systems to be able to position themselves correctly. To solve this problem, the following approach is considered. The solution relies on an IMU to estimate the attitude of the UGV and get a rough estimate of its position by integrating the accelerometer measurements. It also uses the information coming from the wheel encoders to get a better estimate of the position by using a correction algorithm (EKF). The proposed solution can then provide adequate position and attitude estimates.

C. Relevance

The proposed solution is relevant in the way that it is completely on-board. Therefore, it allows the vehicle to operate in environments where global positioning system (GPS) signals are blocked – which can be a key advantage when operating in indoor or underground environments.

D. Scope

As previously mentioned, the proposed solution is entirely based on the article of [1]. However, for academic reasons, the 3D problem is simplified to 2D. Also, only missions of a short period of time are considered allowing to simplify the sensor modeling by neglecting error terms such as scale factors and bias which are commonly present in these types of dead reckoning (DR) sensors.

E. Project Objectives

The first objective of this project is to analyze the kinematics of a four-wheeled and skid-steered vehicle. Then, the second objective is to design an appropriate EKF to estimate its 2D position according to the previously defined scope of the project (see Sec. I-D). Finally, the last objective is to analyze the performance of the developed algorithm in a real-life situation using the Husky UGV made by Clearpath Robotics [2].

II. Proposed Solution

A. Alternative Approaches

Before retaining the solution developed by [1], several other approaches were investigated. Indeed, other approaches exist to solve the positioning

* Presented to Professor Jérôme LE NY to meet the requirements of the *ELE6209A – Navigation Systems* course. Electronically submitted via Moodle on June 26th, 2017.

† Master candidates at Polytechnique Montreal. Contact us: {olivier.gougeon;mohammad-hossein.beheshti}@polymtl.ca

problem. These approaches mainly rely on slow off-board position fixing (PF) systems to correct fast on-board DR systems using a certain fusion algorithm. Some of these combinations include methods like GPS with an inertial navigation system (GPS/INS), light detection and ranging with INS (LiDAR/INS) and vision algorithms with INS (CAM/INS).

B. Chosen Solution

As mentioned in Sec. I-C, the solution using only DR sensors is retained to have an entirely independent and autonomous solution.

DR is one of the major techniques for localization applications and relies on integrating measurements coming from a strapdown IMU to provide rough state estimates using an INS. These first estimates are then corrected using the information coming from two incremental encoders measuring the wheel angular velocities. This solution is made possible since wheel encoders provide information with much more confidence than IMU measurements that are commonly very noisy.

A fusing algorithm similar to the one developed by [1] is then developed to correct the fast IMU readings with the slow encoder readings. An empirical wheel-slip model is also accounted in the correction step using the encoder measurements.

C. Benchmark Mission

A benchmark mission is defined to evaluate the performance of the fusion algorithm developed. Since the biggest part of the modeling error comes from the wheel slip model and that this phenomenon occurs mainly when rotating, a circular path benchmark mission is considered. The defined path is a 20 m circumference circle traveled five times. The total traveled distance is then 100 m and is traveled at a linear tangent velocity of 1 m/s then resulting in a total traveling time of 100 s.

D. Performance Specifications

according to Sec. II-C, the benchmark mission should have a maximum 2D position error of less than 1 m per traveled circle. The total error for five traveled circles should then be less than 5 m for an approximately 3.2 m radius circle.

E. Used Tools

The simulations of the four-wheeled mobile robot are performed using a kinematic model. The continuous time (CT) differential equations are solved using a MATLAB/Simulink environment. In addition, the fusion algorithm is experimentally analyzed using the Husky development platform.

III. Technical Explanations

A. Desired Model States

The desired model states are the controlled ones. Therefore, for the 2D mission proposed in Sec. II-C, the UGV states of interest are the positions (X_I, Y_I, ψ) in the local inertial frame and the velocities (v_{Gx}, v_{Gy}) in the body frame. The local inertial frame is noted as $\mathcal{I}(X, Y, Z)$ and is used to integrate the kinematic equations. The body frame is noted as $\mathcal{B}(x, y, z)$ and is used to develop the vehicle's kinematic. Both frames use the East-North-Up (ENU) right-handed axis system and their orientation is defined in Fig. 2. The parameters and variables are defined in Sec. III-B.1.

B. Studied Platform

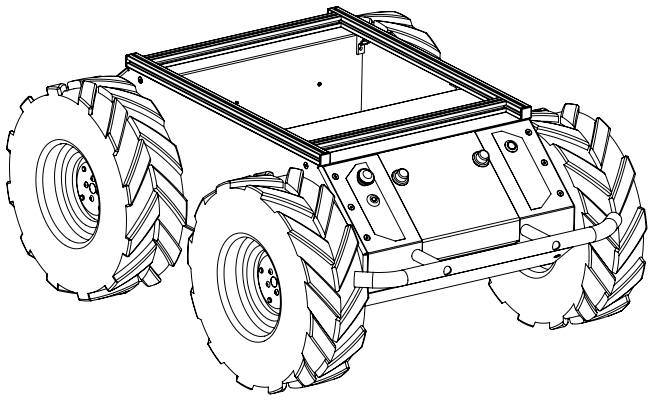

Fig. 1. Husky UGV by Clearpath Robotics [2]

The Husky UGV [2], which is built by Clearpath Robotics, is a medium-sized skid-steered robotic development platform. Four important assumptions are made for its 2D modelization:

1) The center of mass of the UGV is located at its geometric center which is also the origin of the body frame $\mathcal{B}$;
2) The IMU is located at the center of the body frame and its axes are assumed to be aligned with the body frame;

3) Both wheels on each side rotate at the same angular velocity – since they are mechanically locked together;
4) The UGV is always navigating on a flat ground – which states that all four wheels are always in contact with the ground surface that is assumed to be leveled ($\phi = \theta = 0$).

The plant model proposed by [1] is based on the modelization of two distinguishable systems. The first system is the UGV model itself which is based on the kinematic equations of motion for a skid-steered vehicle. The second system is the wheel slip model which is based on an empirical model of the lateral velocity variable S (which is defined in Sec. III-B.2). The latter system is the core of the original contribution of [1] and accounts for all the disturbances that are not modeled in the former system.

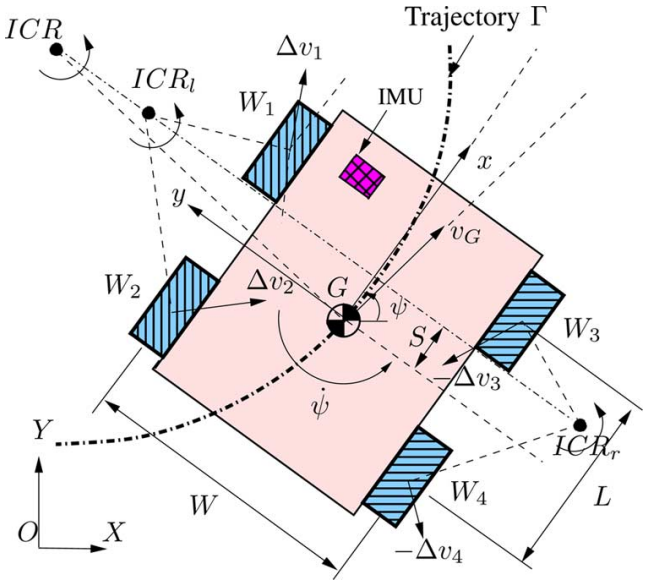

Fig. 2. UGV model parameters used by [1]

1) UGV Model: Based on the previously stated assumptions and the definitions used by [1] (see Fig. 2), we have:

$$\omega_L := \omega_1 = \omega_2 \quad \text{and} \quad \omega_R := \omega_3 = \omega_4 \tag{1}$$

where ω_L and ω_R are respectively the left and right wheel angular velocities and ω_i the wheel angular velocity of wheel i.

Given the wheel radius r, the UGV forward velocity v_{Gx} at its center of mass in frame $\mathcal{B}$, its yaw rate $\dot{\psi}$ and its lateral wheel base W, the left and right wheel forward center velocities are:

$$\begin{aligned} v_{Lx} &= r\omega_L := v_{1x} = v_{2x} = v_{Gx} - \frac{W}{2}\dot{\psi} \\ v_{Rx} &= r\omega_R := v_{3x} = v_{4x} = v_{Gx} + \frac{W}{2}\dot{\psi} \ . \end{aligned} \tag{2}$$

According to the second and fourth assumptions, the IMU velocity vector is given by:

$$\mathbf{v}_B = \begin{bmatrix} v_{Bx} \\ v_{By} \\ v_{Bz} \end{bmatrix} = \begin{bmatrix} v_{Gx} \\ v_{Gy} \\ 0 \end{bmatrix} \ . \tag{3}$$

Finally, the 2D assumption gives:

$$\begin{aligned} \dot{\phi} &= p = 0 \\ \dot{\theta} &= q = 0 \\ \dot{\psi} &= r \ . \end{aligned} \tag{4}$$

Equations (1), (2), (3) and (4) are sufficient to define all the required kinematics to model the motion of the UGV in $\mathcal{B}$. The remaining states of motion in $\mathcal{I}$ can easily be obtained by using the 2D rotation matrix $\mathbf{C}_B^I$ from frames $\mathcal{B}$ to $\mathcal{I}$:

$$\begin{aligned} \mathbf{C}_B^I &= \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \\ \mathbf{V}_I &= \mathbf{C}_B^I \mathbf{v}_B \\ \dot{\mathbf{P}}_I &= \begin{bmatrix} X_I \\ Y_I \end{bmatrix} = \mathbf{V}_I \end{aligned} \tag{5}$$

where $\mathbf{P}_I$ is the UGV position in $\mathcal{I}$. Linear and angular positions in $\mathcal{I}$ are respectively obtained via the integration of (4) and (5) using known initial conditions. The remaining undefined variable v_{Gy} is defined in the following section.

2) Wheel Slip Model: Skid-steered vehicles are special in the way that their wheels *must* slip to change the orientation of the UGV. This characteristic behaviour is very hard to model analytically since in order to obtain the resulting lateral velocity generated during slipping motion, we must estimate the value of the longitudinal instantaneous center of rotation (ICR) location S (see Fig. 2). The authors of [1] prefer using the following

empirical model:

$$\begin{aligned} \gamma &:= \frac{v_{Lx} - v_{Rx}}{v_{Lx} + v_{Rx}} = \frac{\omega_L - \omega_R}{\omega_L + \omega_R} \\ S &:= \frac{a_1}{a_2|\gamma| + a_3} \\ v_{By} &= -S\dot{\psi} = -S\frac{r\,(\omega_R - \omega_L)}{W} \end{aligned} \tag{6}$$

where γ is defined as the ratio of difference and sum of left and right-side's wheel velocities, S is approximated using a curve fit function with coefficients $a_i > 0 \ \forall \ i = 1, 2, 3$ of the collected empirical data and v_{By} is given by its kinematic relationship with respect to S and $\dot{\psi}$.

C. Sensor Theory Overview

In this section, we look at the generic sensor models by considering all the common error sources. Some of these errors are constant, some have known frequency and others are completely random. These errors are common in instruments that are used to calculate the position, the velocity and the attitude of a vehicle.

The IMU gives high rate acceleration, angular rate and magnetic heading measurements in the body frame (second assumption) and the wheel encoders measure the wheels' position and give their angular velocities.

1) Accelerometers: The output of the accelerometers is integrated to calculate the vehicle's position and velocity. Since the acceleration measurements are noisy, their integration will propagate big errors if not corrected. The main errors are classified as the sensor scale factor, a moving bias and a random error. Other error sources such as gravitational, cross-coupling or sensor misalignment errors will not be considered [4].

A model that relates the accelerometer output a_{acc} to the true vehicle acceleration a can be written as:

$$a_{acc} = (1 + SF_{acc})a + b_{acc} + w_{acc}$$

where SF_{acc} and b_{acc} are respectively the accelerometer's scale factor and bias. The wide band sensor noise w_{acc} is considered to be normally distributed with zero mean and variance σ^2_{acc}. We can also consider the moving bias to be normally distributed with zero mean and variance $\sigma^2_{b_{acc}}$.

2) Gyroscopes: The output of the gyroscopes is normally integrated to obtain the Euler angle estimates when initial conditions are known. Since the gyroscope measurements contain different error sources, the calculated attitude angles will also contain errors. They can then be modeled in the same way as the accelerometers:

$$\omega_{gyro} = (1 + SF_{gyro})\omega + b_{gyro} + w_{gyro}$$

where the ω_{gyro} is the reading made by the sensor containing errors, ω is the true angular rate of the vehicle, SF_{gyro} is the scale factor, b_{gyro} is the moving or walking bias and w_{gyro} is the wide band sensor noise [4]. Variances associated to the moving or walking bias $\sigma^2_{b_{gyro}}$ and the wide band sensor noise $\sigma^2_{w_{gyro}}$ can also be considered.

3) Encoders: One of the most familiar used position transducers is the optic incremental encoder which provides electrical pulses while its shaft is rotating. By measuring the rotation of the wheels, we can determine the robot's displacement, velocity or acceleration. It is important to note that optical encoder errors can come from several sources such as quantization error, manufacturing tolerances and ambient effects. These factors can result in erroneous position and velocity readings [5].

The optical encoder will generate one pulse when its shaft rotates from an angle equal to the angular step of graduation θ_p. Therefore, it is clear that the number of pulses N_r generated by the encoder in the course of one complete rotation is:

$$N_r = \frac{2\pi}{\theta_p} \ .$$

Furthermore, the frequency f of the encoder's output is:

$$f = \frac{\omega_r}{2\pi} N_r = \frac{\omega_r}{\theta_p}$$

where ω_r is the angular speed of the shaft. The input signal of the optical encoder is the angular position θ of its shaft with respect to a fixed reference axis. The following definitions are used

to model the encoder outputs:

$$A_{\theta_p}(\theta) = \begin{cases} 1, & 0 \leq \theta < \theta_p/2 \\ 0, & \theta_p/2 \leq \theta < \theta_p \end{cases}$$

$$B_{\theta_p}(\theta) = \begin{cases} 1, & 0 \leq \theta - \theta_p/4 < \theta_p/2 \\ 0, & \theta_p/2 \leq \theta - \theta_p/4 < \theta_p \end{cases}$$

$$Z(\theta) = \begin{cases} 1, & 0 \leq \theta < \theta_p/4 \\ 0, & \theta \geq \theta_p/4 \ . \end{cases}$$

As shown in Fig. 3, the encoder output signals are the two pulses $A(\theta)$ and $B(\theta)$ that are shifted by a quarter angular step ($\theta_p/4$) to identify the direction of rotation. Also, a reference track that has just one window is needed in order to generate a pulse – known as the marker signal $Z(\theta)$ – to initialize the pulse counting of the angular position measurement and to detect each complete revolution. Indeed, it is the combination of these three signals that makes possible the determination of the shaft's position without any ambiguity.

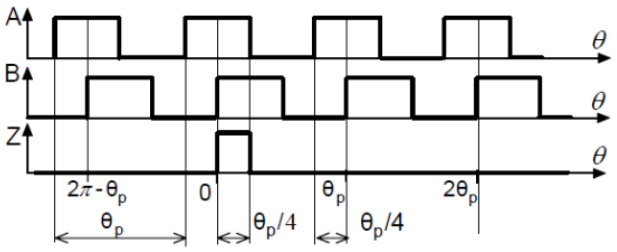

Fig. 3. Output signals generated by the incremental encoder [5]

D. Fusion Algorithm

In Sec. III-B, the UGV and slip models were respectively developed using analytic kinematic equations and empirical data. The equations were parameterized with respect to assumed known variables (such as the linear and angular input velocities) for simulation purposes. However, in real life and in the general case, the value of these variables can't be measured and can only be estimated using fusion algorithms. Therefore, a new kinematic model of the vehicle based on real-life measurements (accelerometers, gyroscopes and encoders) must be defined.

The accelerometers and the gyroscopes respectively measure the linear acceleration $\tilde{\mathbf{A}}_B = \begin{bmatrix} \tilde{a}_{Bx} & \tilde{a}_{By} & \tilde{a}_{Bz} \end{bmatrix}^\top$ and the angular velocity $\tilde{\boldsymbol{\omega}}_B = \begin{bmatrix} \tilde{\omega}_{Bx} & \tilde{\omega}_{By} & \tilde{\omega}_{Bz} \end{bmatrix}^\top$ vectors in $\mathcal{B}$ while the encoders measure the left and right wheel angular velocities $\tilde{\omega}_L$ and $\tilde{\omega}_R$ in their respective frames according to the orientation of the y-axis in $\mathcal{B}$.

Therefore, the problem consists in calculating the best state estimates of the process states presented in Sec. III-B using only these available measurements. It must be specified that the sensors states are not considered in the fusion algorithm retained. As the authors of [1] explain in their article, the wheel slip model combined to the 3D UGV model are sufficient to obtain very good estimations for a short time (a few minutes) robot motions. Indeed, it is mainly this former model that compensates for all the modeling errors, external disturbances and measurement errors. That being said, the sensor models described in Sec. III-C are presented only for completeness considerations.

The algorithm selected for fusing the sensor data is the EKF. The state estimation problem is solved using the same technique as in [6]. In this algorithm, two sorts of information are used:

1) Mathematical models;
2) Sensor measurements.

According to the accuracy of the measurements and the correctness of the 2D model assumptions made in Sec. III-B regarding the UGV and wheel slip models, the EKF will deliver optimal estimates by using recursive integrating and correcting steps such as in [7]. Indeed, it is predicted that lowering the dimension of the general 3D case presented by [1] to the 2D case will have significant impact on the performance of the EKF algorithm in real-life applications when the 2D assumptions are not perfectly met. This matter is intensively discussed in the experimental results analysis of Sec. IV-B.

1) Kinematic Equations of the Process Model: The estimated states in the EKF are the 2D positions and velocities of the IMU in $\mathcal{I}$ and the associated yaw angle of $\mathcal{B}$ relatively to $\mathcal{I}$. The state vector can then be defined as:

$$\mathbf{x} = \begin{bmatrix} X_I & Y_I & V_x & V_y & \psi \end{bmatrix}^\top \ .$$

The kinematic equations associated with these

states are defined as:

$$\begin{aligned}\dot{X}_I &= V_x \\ \dot{Y}_I &= V_y \\ \dot{V}_x &= a_{Bx}\cos\psi - a_{By}\sin\psi \\ \dot{V}_y &= a_{Bx}\sin\psi + a_{By}\cos\psi \\ \dot{\psi} &= \omega_{Bz}\ .\end{aligned} \tag{7}$$

2) Sensor Modeling: The EKF relies on three different kinds of measurements. First, there are the fast DR integrated measurements:

$$\tilde{\boldsymbol{\mu}} = \begin{bmatrix}\tilde{a}_{Bx} & \tilde{a}_{By} & \tilde{\omega}_{Bz}\end{bmatrix}^\top$$

that are integrated at a frequency $f_1 = 100$ Hz in the integration step of the EKF. Then, there are the slow DR correcting measurements:

$$\tilde{\mathbf{y}} = \begin{bmatrix}\tilde{\omega}_L & \tilde{\omega}_R\end{bmatrix}^\top$$

that are being used to calculate the pseudo-correcting measurements:

$$\tilde{\mathbf{y}}^* = \begin{bmatrix}\tilde{v}_{Bx} \\ \tilde{v}_{By}\end{bmatrix} = \begin{bmatrix} {}^{r}/_{2}\left(\tilde{\omega}_R + \tilde{\omega}_L\right) \\ -{}^{rS}/_{W}\left(\tilde{\omega}_R - \tilde{\omega}_L\right)\end{bmatrix} \tag{8}$$

at a frequency $f_2 = 10$ Hz in the correcting step of the EKF. The system (8) is obtained according to the UGV (2) and slip (6) models previously defined in Sec. III-B.

No scale factors or bias are considered in the sensor modeling for the reasons previously mentioned: only additive white Gaussian noise (AWGN) is accounted. We then obtain the following models for the sensors:

$$\begin{aligned}\tilde{a}_{Bx} &= a_{Bx} + \nu_{a_{Bx}} \sim \mathcal{N}(0, \sigma^2_{a_{Bx}}) \\ \tilde{a}_{By} &= a_{By} + \nu_{a_{By}} \sim \mathcal{N}(0, \sigma^2_{a_{By}}) \\ \tilde{\omega}_{Bz} &= \omega_{Bz} + \nu_{\omega_{Bz}} \sim \mathcal{N}(0, \sigma^2_{\omega_{Bz}}) \\ \tilde{\omega}_L &= \omega_L + \nu_{\omega_L} \sim \mathcal{N}(0, \sigma^2_{\omega_L}) \\ \tilde{\omega}_R &= \omega_R + \nu_{\omega_R} \sim \mathcal{N}(0, \sigma^2_{\omega_R})\end{aligned} \tag{9}$$

for which we define the following AWGN vectors respectively associated to f_1 and f_2:

$$\begin{aligned}\mathbf{w} &= \begin{bmatrix}\nu_{a_{Bx}} & \nu_{a_{By}} & \nu_{\omega_{Bz}}\end{bmatrix}^\top \\ \mathbf{v} &= \begin{bmatrix}\nu_{\omega_L} & \nu_{\omega_R}\end{bmatrix}^\top\ .\end{aligned}$$

3) Navigation Equations: The navigation equations of the process and sensor models are given by expressing the system (7) in function of the real measurements given by (9). We obtain:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{w}, t) \\ \dot{X}_I &= V_x \\ \dot{Y}_I &= V_y \\ \dot{V}_x &= \left(\tilde{a}_{Bx} - \nu_{a_{Bx}}\right)\cos\psi - \left(\tilde{a}_{By} - \nu_{a_{By}}\right)\sin\psi \\ \dot{V}_y &= \left(\tilde{a}_{Bx} - \nu_{a_{Bx}}\right)\sin\psi + \left(\tilde{a}_{By} - \nu_{a_{By}}\right)\cos\psi \\ \dot{\psi} &= \tilde{\omega}_{Bz} - \nu_{\omega_{Bz}}\ .\end{aligned}$$

4) Linearized Model: The linearized model is given by the two following Jacobian matrices:

$$\begin{aligned}\mathbf{A}_k &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{(\mathbf{x}=\hat{\mathbf{x}},\mathbf{w}=\mathbf{0},k)} \\ &= \begin{bmatrix}0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -\tilde{a}_{Bx}\sin\hat{\psi} - \tilde{a}_{By}\cos\hat{\psi} \\ 0 & 0 & 0 & 0 & \tilde{a}_{Bx}\cos\hat{\psi} - \tilde{a}_{By}\sin\hat{\psi} \\ 0 & 0 & 0 & 0 & 0\end{bmatrix} \\ \mathbf{B}_k &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{(\mathbf{x}=\hat{\mathbf{x}},\mathbf{w}=\mathbf{0},k)} \\ &= \begin{bmatrix}0 & 0 & 0 \\ 0 & 0 & 0 \\ -\cos\hat{\psi} & \sin\hat{\psi} & 0 \\ -\sin\hat{\psi} & -\cos\hat{\psi} & 0 \\ 0 & 0 & -1\end{bmatrix}\ .\end{aligned}$$

5) Pseudo-Measurement Update Model: The pseudo-measurement update model is defined as:

$$\begin{aligned}\tilde{\mathbf{y}}^* &= \mathbf{h}(\mathbf{x}, \mathbf{v}, k) \\ \tilde{v}_{Bx} &= v_{Bx} + \nu_{v_{Bx}} \sim \mathcal{N}(0, \sigma^2_{v_{Bx}}) \\ &= V_x\cos\psi + V_y\sin\psi + \nu_{v_{Bx}} \\ \tilde{v}_{By} &= v_{By} + \nu_{v_{By}} \sim \mathcal{N}(0, \sigma^2_{v_{By}}) \\ &= -V_x\sin\psi + V_y\cos\psi + \nu_{v_{By}}\end{aligned} \tag{10}$$

for which we obtain the follow two linearized Jacobian matrices:

$$\begin{aligned}\mathbf{C}_k &= \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{(\mathbf{x}=\hat{\mathbf{x}},\mathbf{v}=\mathbf{0},k)} \\ &= \begin{bmatrix}0 & 0 & \mathrm{c}_{\hat{\psi}} & \mathrm{s}_{\hat{\psi}} & -\hat{V}_x\mathrm{s}_{\hat{\psi}} + \hat{V}_y\mathrm{c}_{\hat{\psi}} \\ 0 & 0 & -\mathrm{s}_{\hat{\psi}} & \mathrm{c}_{\hat{\psi}} & -\hat{V}_x\mathrm{c}_{\hat{\psi}} - \hat{V}_y\mathrm{s}_{\hat{\psi}}\end{bmatrix} \\ \mathbf{D}_k &= \left.\frac{\partial \mathbf{h}}{\partial \mathbf{v}}\right|_{(\mathbf{x}=\hat{\mathbf{x}},\mathbf{v}=\mathbf{0},k)} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix}\end{aligned}$$

where $\mathrm{c}_x := \cos x$ and $\mathrm{s}_x := \sin x$.

6) EKF Algorithm: The EKF algorithm is divided in two parts. The first part runs at $f_1 = {}^1/_{\Delta t_1}$ and is used to continuously provide an estimate of the state vector by integrating the fast DR measurements $\tilde{\boldsymbol{\mu}}$. The second part runs at $f_2 = {}^1/_{\Delta t_2}$ and is used to periodically correct the integrated estimates by using the slow DR measurements $\tilde{\mathbf{y}}$.

For the first part, Euler schemes are used to integrate the state vector and the covariance matrix:

$$\begin{aligned}
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \Delta t_1 \mathbf{f}\left(\mathbf{x} = \hat{\mathbf{x}}, \mathbf{w} = \mathbf{0}, k\right) \\
\boldsymbol{\Phi}_k &= e^{\Delta t_1 \mathbf{A}_k} \\
\mathbf{N}_k &= \Delta t_1 \mathbf{B}_k \mathbf{Q} \mathbf{B}_k^\top \\
\boldsymbol{\Sigma}_k &= \boldsymbol{\Phi}_k \boldsymbol{\Sigma}_{k-1} \boldsymbol{\Phi}_k^\top + \mathbf{N}_k
\end{aligned}$$

where $\mathbf{Q}$ is the process noise covariance matrix:

$$\mathbf{Q} = \begin{bmatrix} \sigma^2_{a_{Bx}} & 0 & 0 \\ 0 & \sigma^2_{a_{By}} & 0 \\ 0 & 0 & \sigma^2_{\omega_{Bz}} \end{bmatrix}$$

where $\sigma^2_{acc} = \sigma^2_{a_{Bx}} = \sigma^2_{a_{By}}$ and $\sigma^2_{gyro} = \sigma^2_{\omega_{Bz}}$. For the second part, the following equations are used to correct the current estimates:

$$\begin{aligned}
\mathbf{K}_k &= \boldsymbol{\Sigma}_k \mathbf{C}_k^\top \left(\mathbf{C}_k \boldsymbol{\Sigma}_k \mathbf{C}_k^\top + \mathbf{D}_k \mathbf{R} \mathbf{D}_k^\top\right)^{-1} \\
\hat{\mathbf{x}}_{k,cor} &= \hat{\mathbf{x}}_k + \mathbf{K}_k\big(\tilde{\mathbf{y}}^* - \mathbf{h}(\hat{\mathbf{x}}_k, \mathbf{0}, k)\big) \\
\boldsymbol{\Sigma}_{k,cor} &= \left(\mathbf{I}_5 - \mathbf{K}_k \mathbf{C}_k\right) \boldsymbol{\Sigma}_k
\end{aligned}$$

where $\mathbf{R}$ is the measurement noise covariance matrix:

$$\mathbf{R} = \begin{bmatrix} \sigma^2_{v_{Bx}} & 0 \\ 0 & \sigma^2_{v_{By}} \end{bmatrix} = \begin{bmatrix} r^2\sigma^2_{enc} & 0 \\ 0 & \alpha^2\sigma^2_{v_{Bx}} \end{bmatrix}$$

with $\sigma^2_{enc} = \sigma^2_{\omega_L} = \sigma^2_{\omega_R}$ and α a tuning parameter representing the relative confidence between the UGV and slip models.

The EKF algorithm has the following scheme:

1) Initialize the integration first step with $\hat{\mathbf{x}}_0$ and $\boldsymbol{\Sigma}_0$;
2) Integrate $\hat{\mathbf{x}}_{k-1}, \boldsymbol{\Sigma}_{k-1}$ to obtain $\hat{\mathbf{x}}_k, \boldsymbol{\Sigma}_k$;
3) If $\tilde{\mathbf{y}}$ is available: correct $\hat{\mathbf{x}}_k, \boldsymbol{\Sigma}_k$; otherwise: keep integrating.

IV. Results

A. Simulation Results

To analyze the performance of the proposed sensor fusion algorithm, two tests are performed on the 2D model of the Husky platform. The first test consists in submitting the EKF algorithm to the circular reference trajectory previously defined in II-C. The second test consists in evaluating the modified mean squared error (MMSE) of the estimated $\hat{X}_I$ and $\hat{Y}_I$ position states by running several Monte Carlo simulations until convergence is reached with an acceptable tolerance.

TABLE I
INITIALIZATION PARAMETERS USED FOR THE SIMULATIONS

Description	Symbol	Value	Units
Longitudinal wheel base	W	0.555	m
Effective wheel radius	r	0.165	m
First wheel slip coefficient	a_1	0.02148	m
Second wheel slip coefficient	a_2	0.249	ratio
Third wheel slip coefficient	a_3	0.039	ratio
Desired forward velocity	v_{Gx}	1	m/s
Desired turning rate	$\dot{\psi}$	0.3142	rad/s

TABLE II
SENSOR PARAMETERS USED FOR THE SIMULATIONS

Description	Symbol	Value	Units
Accelerometer PSD	σ^2_{acc}	0.0008^2	$(\mathrm{m/s^2})^2/\mathrm{Hz}$
Gyroscope PSD	σ^2_{gyro}	0.0005^2	$(\mathrm{rad/s})^2/\mathrm{Hz}$
Encoder PSD	σ^2_{enc}	0.0001^2	$(\mathrm{rad/s})^2/\mathrm{Hz}$
Confidence parameter	α^2	2^2	ratio

The parameters used in the simulations are presented in Tab. I and II which respectively present the initialization and sensor parameters. Zero value initial position states are considered: $X_{I_0} = Y_{I_0} = \psi_0 = 0$. The initial velocity states V_{x_0} and V_{y_0} are calculated using the 2D rotation matrix $\mathbf{C}_B^I$ with respect to the desired forward and turning velocities defined in Tab. I. For the EKF algorithm, the initial estimated states are assumed to be precisely known ($\hat{\mathbf{x}}_0 = \mathbf{x}_0$) and therefore a very small covariance matrix is used ($\boldsymbol{\Sigma}_0 = 0.0001^2\mathbf{I}_5$). The accelerometer and gyroscope power spectrum density (PSD) values (respectively σ^2_{acc} and σ^2_{gyro}) are the ones associated to the AHRS installed in the Husky. They were found in the datasheet of the manufacturer [8]. For simplified modeling purposes, a much smaller PSD value was assigned

to the encoder PSD (σ^2_{enc}) since it is commonly much more precise than an accelerometer or a gyroscope. Finally, a relative confidence tuning parameter α greater than one is used to simulate the fact that lower confidence is attributed to the slip model than to the UGV model.

1) Benchmark Test: According to the simulation parameters previously defined, the benchmark test is conducted. Fig. 4 and 5 show the obtained results for the circular path traveled five times for one specific realization of the random parameters.

Fig. 4 shows the comparison between the reference and integrated paths. The integrated path corresponds to the $\hat{X}_I$ and $\hat{Y}_I$ position estimates given by the EKF algorithm when applying *only* its integrating part. The result shows that, when using only the integration of IMU measurements, the position estimates of the UGV are acceptable during the first lap. However, we observe that the estimated coordinates quickly diverge from the actual path as the time-integration process goes on. The *only* integration approach without using the encoder sensor updates diverge as the error covariance matrix increases. As a matter of fact, $\mathbf{\Sigma}_k$ quickly explodes to `NaN` values.

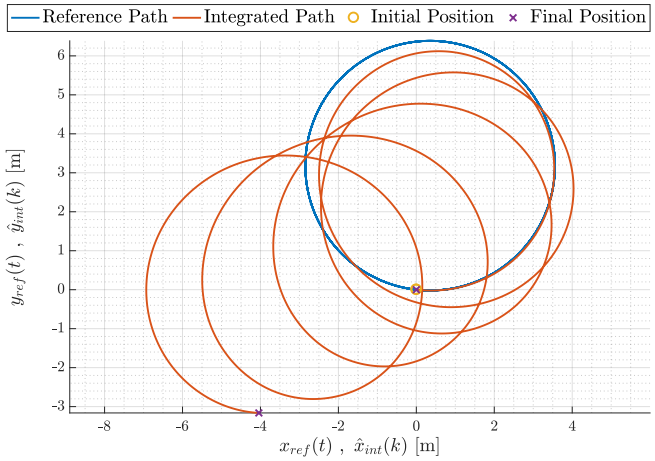

Fig. 4. Comparison between the reference and integrated paths of the simulation results

Fig. 5 shows the comparison between the reference and estimated paths. The estimated path corresponds to the $\hat{X}_I$ and $\hat{Y}_I$ position estimates given by the EKF algorithm when applying *both* its integrating and correcting parts. The result shows that, when using only the integration of IMU measurements combined to the correction given by the pseudo-measurements, the position estimates of the UGV are much better. Indeed, a good performance is observed with the EKF algorithm since the estimation follows the reference path and the estimation does not diverge due the encoder speed correction. Therefore, this result shows an improved estimation and very acceptable performance – without the need of using PF systems that are commonly not available when working in indoor environments. However, we note that the corrected estimates do tend to diverge a little in a certain direction. This phenomenon is due to the fact that the correction step is working at a frequency ten times lower than the integration step. It is then impossible to account for *all* of the integration errors.

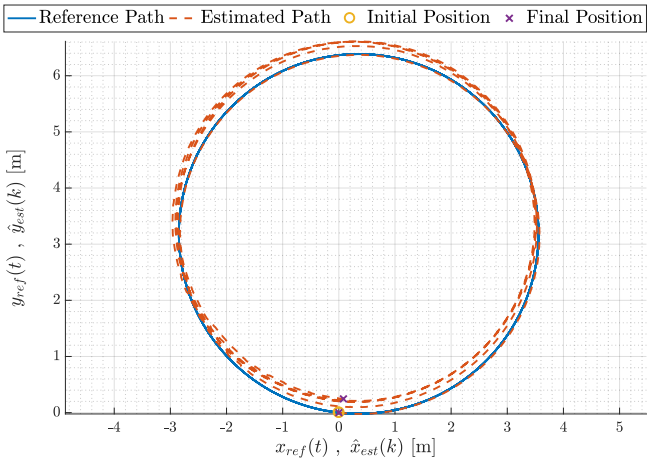

Fig. 5. Comparison between the reference and estimated paths of the simulation results

2) Performance Analysis: The performance analysis will be conducted using the modified mean squared error (MMSE) measurement for time-based simulations:

$$\text{MMSE}(\hat{x}) = \frac{1}{N}\sum_{i=1}^{N}\int_0^T \left|\hat{x}_i(t) - x_i(t)\right|^2 dt$$

where N is the number of Monte Carlo simulations, T is the time of each simulation, $\hat{x}_i$ and x_i are respectively the EKF estimated and true plant analyzed states ($\hat{X}_I$ and $\hat{Y}_I$) for each Gaussian random case i analyzed.

For this analysis, $N = 34\ 000$ Monte Carlo simulations were realized for an individual true simulation time of $T = 5 \times 20 = 100$ s. Fig. 6

shows the evolution of the recursive mean of the MMSE measurement calculations for $\hat{X}_I$ and $\hat{Y}_I$ in function of the number of Monte Carlo simulations. It must be noted that *not* all the iteration results were plotted. However, the results show that the MMSE values tend to change a lot at the beginning of the simulations and converge to a fixed value as i grows. The final obtained values are $\text{MMSE}(\hat{X}_I) = 1.6785$ m^{2}s and $\text{MMSE}(\hat{Y}_I) = 2.4855$ m^{2}s. If we retain only the worst case, this means that after 100 s (five circular laps) the maximum position error will have a standard deviation of $\sqrt{2.4855} = 1.5765$ m$\sqrt{\text{s}}$ which is much smaller than the initial performance specifications mentioned in Sec. II-D (i.e. 5 m).

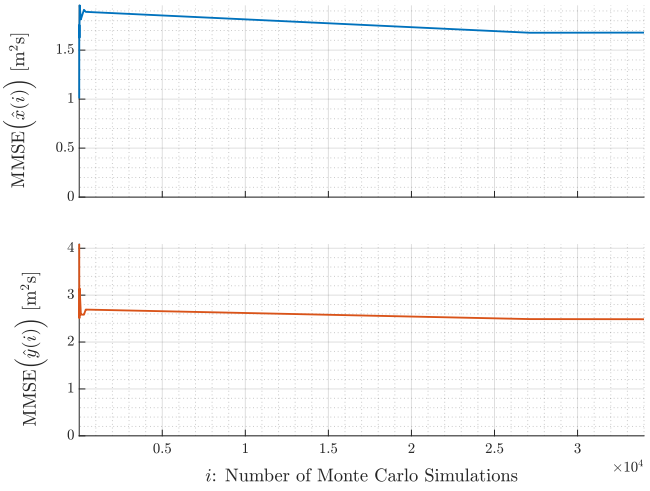

Fig. 6. MMSE of the $\hat{X}_I$ and $\hat{Y}_I$ estimated coordinates of the simulated path

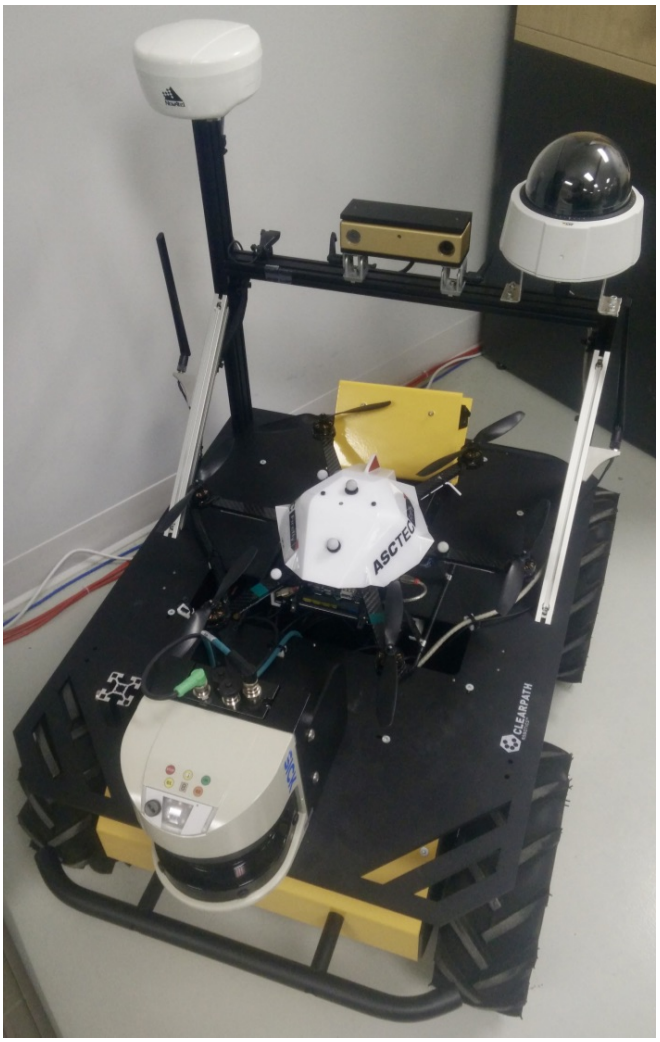

Fig. 7. Experimental Husky configuration

B. Experimental Results

Experimental data is gathered to make a comparison with the simulated results. The objective of the experiment is to evaluate the real-life performance of the 2D EKF algorithm developed in indoor environments. The sensor measurements are collected in real-time from the Husky AHRS and encoders. They are then processed offline with the MATLAB software to inject them in the EKF algorithm. The reference 2D positions of the UGV are collected via the VICON motion capture system that uses infrared cameras which acts as a PF system to get the "true" path. Fig. 7 shows the experimental Husky configuration. The AscTec Firefly hexacopter was temporary fixed at the center of gravity location to provide quick and easy data acquisition via the VICON system. The experiment was conducted on June 25th of 2017 at Polytechnique Montréal in the Mobile Robotics and Autonomous Systems Laboratory (MRASL). A high definition video of the conducted experiment is available on Dropbox [9].

The experimental results are presented in Fig. 8 which shows a comparison between the reference and estimated paths. The estimated path is obtained using the exact same EKF algorithm used in the simulations (i.e. only the encoders are used for the correction step although PF data is available via the VICON system). The results demonstrate that the VICON system gives very good position readings (millimeter accuracy when well calibrated).

Without any surprise, the analysis of Fig. 8

shows that the simulation model achieved better DR estimation accuracy than when experimental data was used in the EKF. Generally, from a qualitative standpoint, the experimental results are coherent with those obtained in the simulated model. However, the experimental path error is much bigger than the simulated one because of uncertain model parameters, complex wheel/ground interactions and kinematic constraints. Furthermore, these results confirm that the 2D hypothesis (i.e. a perfectly leveled surface) is too strong to be valid and that one must account the full 3D kinematic of the vehicle to obtain acceptable estimates. Another important source of error is the misalignment of the AHRS frame with respect to the body frame of the vehicle. Indeed, these two sources of error will induce important errors in the $\tilde{a}_{Bx}$ and $\tilde{a}_{By}$ acceleration measurements that will then be reading some vector components of the gravity vector g. Only a few degrees of error will then produce important errors in the double integration process to obtain position estimates.

It must also be said that the comparison between the reference states and the *only* integrated estimates (i.e. without correction) is not shown because the integrated states quickly diverge due to many real-life errors not accounted for in the 2D model as just previously explained.

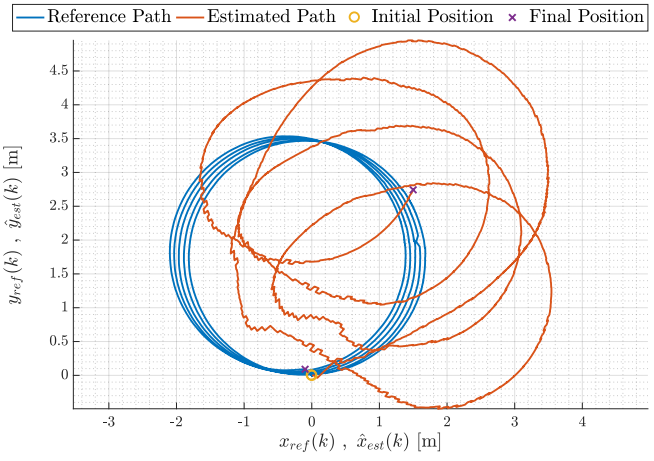

Fig. 8. Comparison between the reference and estimated paths of the experimental results

V. Conclusion

A. Completed Work Overview

A kinematic model of a skid-steered UGV was developed based on [1]. Also, based on the IMU and wheel encoder measurements, an efficient EKF fusion framework was designed throughout many simulations and the algorithm was tested experimentally on the Husky platform. The simulation results showed satisfying performances for the proposed EKF methodology. However, the experimental results showed that the 2D model assumptions are not valid and that the more general 3D case must be considered for real-life applications.

B. Personal Evaluation

ELE6209A Project Timeline		**May 8**	**May 11**	**May 15**	**May 18**	**May 22**	**May 25**	**May 29**	**Jun 1**	**Jun 5**	**Jun 8**	**Jun 12**	**Jun 15**	**Jun 19**	**Jun 22**	**Jun 26**
#	**Step**	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B	7A	7B	8A
0	Team formation															
0	Project Definition															
0	Project Proposal															
1	Reading															
2	Plant Modelling															
3	Sensors Modelling															
4	Sensor Fusion															
5	Simulation															
0	Progress Report															
6	Performance Analysis															
7	Real Life Implementation															
8	Real Life Perf. Analysis															
9	Writing															
0	Project Presentation															
0	Final Report															

Fig. 9. Initial proposed timeline presented in a Gantt chart

The proposed fusion algorithm gave good simulation results for the 2D vehicle model and was validated on a circular path. It was shown that the proposed EKF improves the system performance in terms of reduced DR error when using the correction provided by the wheel encoders. However, the proposed solution remains weak for real-life applications and the full 3D case must be accounted for.

The initial project proposal stated that the full 3D model would be studied during the project. However, because of time constraints, the 2D model was preferred to make possible the comparison with experimental data – which was an important objective for our team.

Finally, because of the rapidity of the summer courses schedule and the workload associated with the course, it was not possible to better investigate the real-life experimental case. Fig. 9 shows the initial proposed timeline that was followed during the project. It can be said that not enough time was planned for the simulation part of the EKF since many problems surged during this phase.

C. Future Works

It is clear that the simulation model achieved better estimation accuracy than the experimental results and that the 2D considered model is the main failure cause for the real-life implementation. Therefore, we propose that the full 3D UGV modeling and fusion estimation algorithm be considered for future works. Finally, after providing a solid solution to the navigation problem, it is suggested that the guidance and control problems be also investigated as interesting subjects for research and development considerations.

D. Copyright

The authors of this report allow anyone to reuse the material that was produced in the context of this project for *educational purposes only*. Proper acknowledgment of the work must be done (i.e. using citations and references). Indeed, some material such as the step reports, the adapted method from [1], the simulation files and results and the experimental data gathered with the Husky and VICON systems could be of high interest for anyone who wants to push the analysis of the studied problem throughout this project.

ACKNOWLEDGEMENT

The authors of this report also acknowledge the help that was provided by André PHU-VAN NGUYEN, Quang NGUYEN DANG and David ST-ONGE – without any order of preference – for the experimental part of this project.

REFERENCES

[1] J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya, and J. Liu, "Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation," *IEEE transactions on robotics*, vol. 25, no. 5, pp. 1087–1097, 2009.

[2] Clearpath Robotics, "Husky Unmanned Ground Vehicle," May 2017. [Online]. Available: https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

[3] M. Hennessey, "Robots explore dangerous mines with novel sensor fusion technology," Clearpath Robotics, Jan. 2015. [Online]. Available: https://www.clearpathrobotics.com/2015/01/robots-explore-dangerous-mines-sensor-fusion-technology/

[4] D. Gebre-Egziabher, "Design and performance analysis of a low-cost aided dead reckoning navigator," *PhD Thsis, Stanford University*, 2004.

[5] I. I. Incze, C. Szabó, and M. Imecs, "Incremental encoder in electrical drives: modeling and simulation," in *Computational Intelligence in Engineering*. Springer, 2010, pp. 287–300.

[6] K. Berntorp, K.-E. Årzén, and A. Robertsson, "Sensor fusion for motion estimation of mobile robots with compensation for out-of-sequence measurements," in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*. IEEE, 2011, pp. 211–216.

[7] K. Gade, "Introduction to inertial navigation and kalman filtering," in *Tutorial for IAIN World Congress, Stockholm*, 2009, pp. 1–57.

[8] LORD Corporation, "3DM-GX3-25 AHRS Datasheet," 2014, LORD Sensing MicroStrain. [Online]. Available: http://files.microstrain.com/3DM-GX3-25-Attitude-Heading-Reference-System-Data-Sheet.pdf

[9] O. Gougeon, "Experimental Data Acquisition with the Husky UGV," Video, Jun. 2017, Polytechnique Montréal. [Online]. Available: https://www.dropbox.com/s/tr61h6dyvaxcxvk/Experimental-Data-Acquisition-2017-06-25-02-12-55.mov?dl=0